




I. Objectif du TP :

Objectif du TP est d'afficher la liste des étudiants réussis avec la mention à partir d'un fichier xml qui contient 4 étudiants, des étudiants réussis et des étudiants non réussis avec des avec différent mensions.

 Rapport_XML_XSL.docx	07/05/2021 2:37
 liste.xml	06/05/2021 23:39
 liste.xsl	06/05/2021 23:40



II. Définition de l' XSL :

XSL (eXtensible Stylesheet Language) est le langage de description de feuilles de style du W3C associé à XML.

Une feuille de style XSL est un fichier qui décrit comment doivent être transformés les documents XML basés sur une même DTD ou un même schéma.

La spécification est divisée en trois parties :

XSLT, le langage de transformation

XPath, le langage de navigation dans un document XML

XSL-FO, le vocabulaire XML de mise en forme

III. Le code :

1. Le fichier XML :

```
<?xml version ="1.0" ?>
<?xml-stylesheet type="text/xsl" href="liste.xsl" ?>

<liste>

  <étudiant code="E1">
    <nom>Moujtahid Moujidd</nom>
    <moyenne>18.91</moyenne>
  </étudiant>

  <étudiant code="E2">
    <nom>Kaddouri Kaddour</nom>
    <tél>070333333</tél>
    <moyenne>15.90</moyenne>
  </étudiant>

  <étudiant code="E3">
    <nom>Jallouli Jalloul</nom>
    <tél>035000002</tél>
    <tél>067000002</tél>
    <moyenne>13.50</moyenne>
  </étudiant>

  <étudiant code="E4">
    <nom>Kaslani Kassoul</nom>
    <tél>035444444</tél>
    <tél>063444444</tél>
    <tél>067444444</tél>
    <moyenne>3.50</moyenne>
  </étudiant>

</liste>
```

Le fichier xml contient 4 étudiants, des étudiants réussis et des étudiants non réussis avec des avec différent mensions.

2. Le fichier XSL :

```
<?xml version='1.0'?>

<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

<xsl:template match="/">

    <html>
    <body>
        <h1>Liste des étudiants réussis</h1>
    <hr />

    <ol>
    <xsl:for-each select="/liste/étudiant[moyenne >= 10]">
    <xsl:sort select="nom" data-type="text" order="descending" />

    <!--xsl:if test="moyenne >= 10" -->
        <li>
            <xsl:value-of select="@code" />--- <xsl:value-of select="nom" />

            (
                <xsl:choose>
                    <xsl:when test="moyenne > 16 ">
                        Très Bien
                    </xsl:when>

                    <xsl:when test="moyenne >= 14 and moyenne < 16">
                        Bien
                    </xsl:when>

                    <xsl:when test="moyenne >= 12 and moyenne < 14">
                        Assez Bien
                    </xsl:when>

                    <xsl:when test="moyenne >= 10 and moyenne < 12">
                        Passable
                    </xsl:when>
                )
            )
        </li>
    </xsl:for-each>
    </ol>
    </body>
    </html>
</template>
</xsl:stylesheet>
```

```

        <xsl:otherwise>
            Ajourné
        </xsl:otherwise>
    </xsl:choose>

)

</li>
<!--/xsl:if-->
</xsl:for-each>
</ol>

</body>
</html>
</xsl:template>

</xsl:stylesheet><!-- Stylus :
<metaInformation>
<scenarios ><scenario default=
</metaInformation>
-->

```

Le fichier xsl contient le code pour transformer le fichier xml à un fichier qui affiche la liste des étudiants réussis avec la mention à partir d'un fichier xml.

IV. Exécution :

1. Exécution du fichier « liste.xsl » :

```

x Validating liste.xsl...
! The XML file does not have a schema to validate against.
!

```

```
<?xml version="1.0"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
<!-- Stylus Studio meta-information - (c) 2004-2006, Progress Software Corporation. All rights reserved. <metaInformation> <scenarios> <scenario default="yes" name="Scenario1"
userelativepaths="yes" externalpreview="no" url="Liste.xml" htmlbaseuri="" outputuri=".\\XSL_2017_2\\liste.html" processortype="internal" useresolver="yes" profilemode="0"
profiledepth="" profilelength="" urlprofilexml="" commandline="" additionalpath="" additionalclasspath="" postprocessortype="none" postprocesscommandline=""
postprocessadditionalpath="" postprocessgeneratedext="" validateoutput="no" validator="internal" customvalidator=""> </scenarios> <MapperMetaTag> <MapperInfo
srcSchemaPathIsRelative="yes" srcSchemaInterpretAsXML="no" destSchemaPath="" destSchemaRoot="" destSchemaPathIsRelative="yes"
destSchemaInterpretAsXML="no"/> </MapperBlockPosition> </MapperBlockPosition> <TemplateContext> </TemplateContext> <MapperFilter side="source"> </MapperFilter> </MapperMetaTag>
</metaInformation> -->
- <xsl:template match="/">
- <html>
- <body>
<h1>Liste des étudiants réussis</h1>
<hr/>
- <ol>
- <xsl:for-each select="/liste/étudiant[moyenne >= 10]">
<xsl:sort select="nom" order="descending" data-type="text"/>
<!--xsl:if test="moyenne >= 10" -->
- <li>
<xsl:value-of select="@code"/>
)---
<xsl:value-of select="nom"/>
(
- <xsl:choose>
<xsl:when test="moyenne > 16"> Très Bien </xsl:when>
<xsl:when test="moyenne >= 14 and moyenne < 16"> Bien </xsl:when>
<xsl:when test="moyenne >= 12 and moyenne < 14"> Assez Bien </xsl:when>
<xsl:when test="moyenne >= 10 and moyenne < 12"> Passable </xsl:when>
<xsl:otherwise> Ajourné </xsl:otherwise>
</xsl:choose>
)
</li>
<!--/xsl:if-->
</xsl:for-each>
</ol>
</body>
</html>
</xsl:template>
</xsl:stylesheet>
```

2. Exécution du fichier « liste.xml » :

Liste des étudiants réussis

1. E1)--- Moujtahid Moujidd (Très Bien)
2. E2)--- Kaddouri Kaddour (Bien)
3. E3)--- Jallouli Jalloul (Assez Bien)

```
x Validating liste.xml...
^ The XML file does not have a schema to validate against.
^
```



Liste des étudiants réussis

1. E1)--- Moujtahid Moujidd (Très Bien)
2. E2)--- Kaddouri Kaddour (Bien)
3. E3)--- Jallouli Jalloul (Assez Bien)

V. Remarquer :

Le **XSL** ne permet pas uniquement l'affichage de XML. Il permet aussi : de sélectionner une partie des éléments XML. de trier des éléments XML.